ENGINEERING

Gestione Utenti Jina

Società	*Engineering*
Area Applicativa:	
Redatto da:	*Tiziana Franchi – Antonio Giglio*
Verificato da:	
Approvato da:	
Numero pagine totale	

VERSIONI

Versione	Motivazione	Data Rilascio
1.0	Nuova gestione utenti Jina	Gennaio-2025
1.1	Inserimento paragrafi tecnici di Antonio Giglio	Marzo-2025

ENGINEERING

Sommario

1. SCOPO E CONTENUTI DEL DOCUMENTO

Il presente documento partendo dalle indicazioni fornite nel capitolo 4.6 del **Manuale di Amministrazione EESSI - JINA 6.2.18 (Portal 6.2.19)** e dal documento **EESSI - JINA - Identity and Access Management (IAM) - rev02**, analizza e propone una nuova gestione delle identità e degli accessi, un aspetto cruciale per garantire la sicurezza e l'affidabilità delle operazioni effettuate attraverso il sistema JINA.

2. DESCRIZIONE DELLA PROCEDURA

La funzionalità di Identity and Access Management (IAM) consente di configurare utenti e gruppi in JINA.

IAM in JINA serve come sistema centrale per controllare chi può accedere a quali funzionalità o dati. Le configurazioni principali includono:

1. **Gestione utenti**:
    - Creazione, modifica ed eliminazione degli utenti.
    - Configurazione di ruoli e appartenenze ai gruppi.
2. **Gestione gruppi**:
    - Organizzazione degli utenti in gruppi logici.
    - Assegnazione di permessi a livello di gruppo per semplificare la gestione.

IAM utilizza un modello multi-tenant, in cui ogni tenant può avere configurazioni uniche, garantendo flessibilità e sicurezza. Gli amministratori possono assegnare permessi a livello di utente o gruppo, ma l'attuale utilizzo si rifà direttamente ai gruppi.

2.1. POLICY

In Jina, come in tutti i sistemi informatici, una **policy** è un insieme di regole e configurazioni che definiscono il comportamento o i permessi all'interno del sistema, di utenti, o gruppo di essi. Le policy servono a garantire il controllo, la sicurezza, ma anche per mantenere la conformità.

Le policy sono generalmente costituite da:

- **Regole**: I criteri specifici che determinano il comportamento (es. "un utente con il ruolo di Supervisor può accedere a tutte le funzionalità").
- **Ruoli**: I soggetti (utenti o gruppi) a cui si applicano le regole.
- **Condizioni**: Eventuali circostanze che devono essere soddisfatte per applicare la policy

2.2. RUOLI

Il sistema consente di assegnare ruoli e autorizzazioni specifiche a utenti e gruppi per garantire un controllo granulare delle operazioni consentite.
Ruoli utente di JINA:

1. **Supervisor**: ha accesso completo a molteplici azioni di gestione.
2. **Authorised e Unauthorised**: utenti con permessi limitati per creare e gestire casi.
3. **Auditor, Viewer, Medical, VIP**: ruoli con autorizzazioni specifiche per la visualizzazione o gestione di dati sensibili.
4. **Everyone**: gruppo generico con permessi minimi.

RINA USER ROLES		Supervisor	Authorised	Unauthorised	Auditor	Viewer	Medical	VIP	Everyone
Case Assignment	Assign Case	✓							
	Request Assignment	✓	✓	✓	✓	✓	✓	✓	✓
	Authorise Assignment	✓							
Case Management	Create Case		✓	✓					
	Create SED		✓	✓					
	View SED	✓	✓	✓	✓	✓			
	Send SED	✓	✓						
	Request Approval for sending			✓					
	Approve and Send SED	✓	✓						
	View/Upload/ Delete/ Send/Download medical attachments						✓		
	View VIPS							✓	
	Change Case Metadata *	✓	✓	✓					
	Manual case archiving	✓	✓						
	Manual case unarchiving	✓	✓						
	Set/clear alarms	✓	✓	✓					
	View case metadata*	✓	✓	✓	✓	✓	✓	✓	✓
	View/download/send regular attachments	✓	✓	✓	✓	✓			
	Add-Upload/delete regular attachments		✓	✓					
	Add/delete comments	✓	✓	✓	✓	✓			
Audit	View Audit	✓			✓				

3. NECESSITÀ DI REVISIONE – MOTIVI E BENEFICI

Come detto IAM (Identity and Access Management) gestisce le identità e gli accessi al sistema JINA. I tre cardini su si poggia questa gestione sono:

- La gestione degli utenti e dei gruppi;
- Le Policy;
- **Le modalità di autenticazione e autorizzazione.**

Proprio quest'ultimo punto presenta molte criticità, e la sua revisione è l'obiettivo primario di questa analisi e documentazione.

3.1. LA MODALITÀ DI AUTENTICAZIONE E AUTORIZZAZIONE

Attualmente questa fase evidenzia alcune criticità nelle modalità adottate e si rende necessaria una revisione approfondita delle procedure esistenti per migliorare l'efficienza e ridurre la complessità operativa.

Il sistema prevede due principali modalità per l'inserimento degli utenti:

1. Inserimento manuale: Questo metodo è oneroso in termini di tempo e soggetto a errori umani, soprattutto quando si gestisce un numero elevato di utenti.
2. Utilizzo di un batch: Richiede la preparazione preventiva di un file Excel, che deve essere fornito come input al sistema. Tuttavia, la produzione di questo file è interamente manuale, il che introduce ulteriori complessità e potenziali fonti di errore.

Le criticità principali derivano dal fatto che:

- Informazioni parcellizzate: I dati relativi ai nuovi utenti da inserire vengono spesso ricevuti in maniera frammentaria e disorganizzata.
- Necessità di formattazione manuale: I dati ricevuti devono essere elaborati e formattati manualmente per adattarsi ai requisiti del file Excel richiesto dal batch. Questo processo non solo aumenta il carico di lavoro, ma può anche portare a incongruenze nei dati inseriti.

Alla luce di queste problematiche, risulta essenziale analizzare e ripensare queste modalità operative per adottare un approccio automatizzato e autonomo, meno soggetto a errori, garantendo al contempo una maggiore scalabilità e semplicità nella gestione delle identità e degli accessi.

Requisito cardine di ogni nuovo inserimento e/o modifica/cancellazione di un utente è emissione di un **Ordine di Servizio.**

4. ATTUALE FUNZIONAMENTO DEL BATCH

Per attivare/lanciare il processo Batch occorre potersi collegare con i server virtuali Jina in rete INPS, in modo pratico ciò attualmente avviene collegandosi in Remote Desktop ad una macchina INPS.
Il nome della procedura è: RinaConfigurator_V2.0.
Le directory interessate alla procedura sono le seguenti:

- RinaConfigurator_V2.0
  - Bin
  - Executor
  - Ln

Nell'ultima directory devo essere inseriti i files di input, sempre nella stessa directory verranno scritti gli output della procedura (vedi paragrafo 4.1).

4.1. FILES DI INPUT/OUTPUT DELLA PROCEDURA BATCH

I files in input alla procedura batch devono avere dei nomi standard al fine di essere riconosciuti ed elaborati dal batch. Le tre nomenclature da seguire sono:

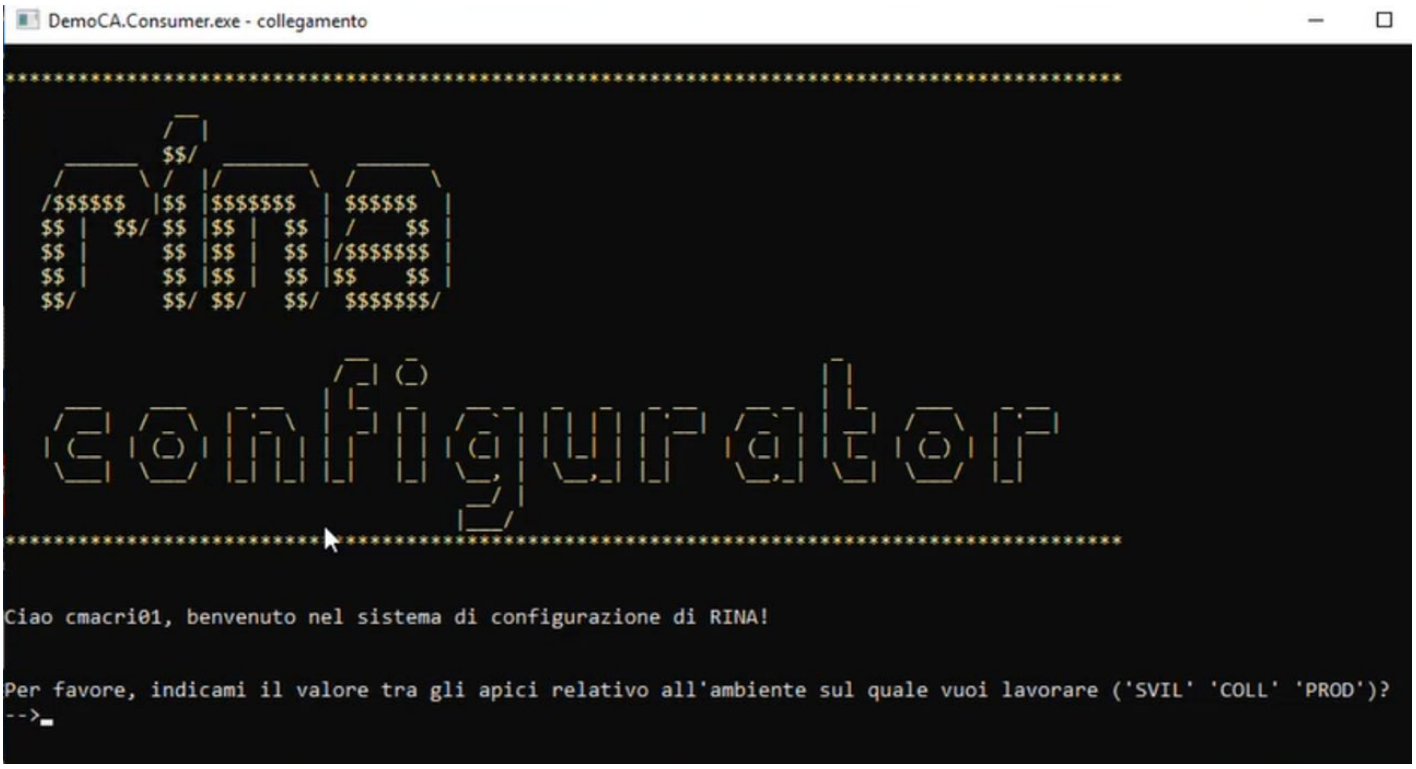
- USERS se si vogliono inserire e/o modificare degli utenti;
- GROUPS se si vogliono inserire e/o modificare dei gruppi;
- POLICIES se si vogliono inserire e/o modificare delle policies;

Il file di input **USERS** segue il formato di testo di un file Excel che l'operatore compila manualmente, con i seguenti campi:

- Cognome
- Nome
- Sede
- Recapito Telefonico
- Email
- Utenza
- PW
- CP
- CO
- Viewer
- Supervisor
- Authorized Clerk
- Medical User
- Un-Authorized Clerk
- VIP
- BUC Sector

4.2. ESECUZIONE DELLA PROCEDURA BATCH

Con il lancio dell'eseguibile: DemoCA.Consumer.exe (….RinaConfigurator_V2.0\Executor\bin) viene avviata la procedura Rina Configurator



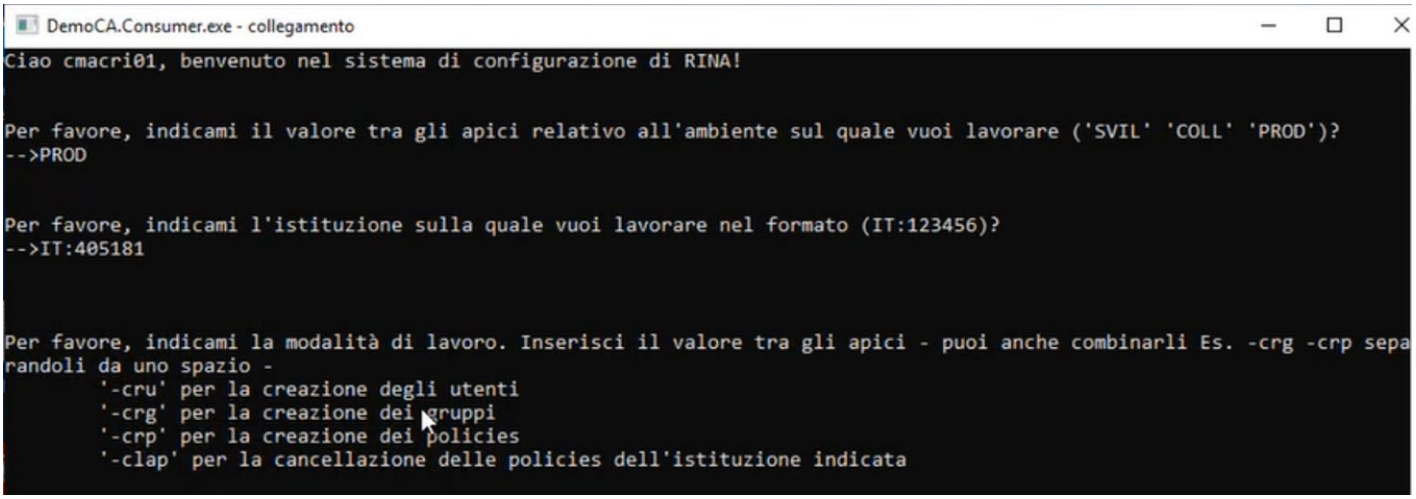
Per la sua corretta esecuzione Rina Configurator necessittà delle seguenti informazioni:

1. Ambiente di lavoro:
   - SVIL
   - COLL
   - PROD

2. Tenant (istituzione di riferimento) nel formato IT:999999

3. Modalità di lavoro:
   - -CRU creazione utenti
   - -CRG creazione gruppi
   - -CRP creazione policies
   - -CLAP cancellazione delle policies

4. File di input (come da paragrafo 4.1)



4.3. SINTESI DEL FLUSSO DEL PROGRAMMA

In questo paragrafo verrà presentata una panoramica del flusso di esecuzione del programma, evidenziando le fasi principali e le interazioni tra i vari moduli.

1. Inizializzazione
   Legge le configurazioni dai file.

2. Avvio della Ricerca File (StartSearch)
   Cerca file con un pattern specificato. Se trova file, aggiorna i percorsi delle directory di lavoro.

3. Controllo delle Cartelle di Destinazione (CheckTargetFolder)
   Verifica la presenza e l'accessibilità delle cartelle di destinazione.

4. Gestione Modalità di Esecuzione
   Modalità interattiva: Se non ci sono argomenti, avvia la modalità interattiva. Modalità parametrica: Altrimenti, legge i parametri per l'ambiente e le modalità operative.

5. Azioni in Base alle Modalità Scelte
   Aggiornamento repository: (-a, -as)
   Cancellazione utenti: (-clf, -cla)
   Creazione utenti: (-cra, -cru)

6. Log e Chiusura
   Registra le operazioni e chiude l'esecuzione.

4.4. DETTAGLIO TECNICO DEL PROGRAMMA

Questo documento fornisce il dettaglio del flusso di cancellazione e inserimento degli utenti dell'applicazione nel sistema **RINA**.

Le principali classi esaminate sono:

- **Program.cs**: Punto di ingresso dell'applicazione che gestisce l'interfaccia utente e il flusso di esecuzione.
- **ActionManager.cs**: Classe responsabile delle operazioni principali che gestisce anche la creazione e la cancellazione di utenti.
- **AuthenticationManager**: Classe che gestisce l'autenticazione con il sistema JINA, ottenendo i token necessari per le comunicazioni successive.

L'analisi integra ulteriori dettagli operativi, in particolare sulle procedure di autenticazione e sulla gestione delle modalità di esecuzione.

4.5. ARCHITETTURA GENERALE

L'applicazione è suddivisa in tre componenti principali:

1. **DemoCA.Consumer** (Program.cs): Gestisce l'interfaccia con l'utente, l'interpretazione dei comandi e l'inizializzazione delle operazioni.
2. **DemoCA.Service.Workflow** (ActionManager.cs): Contiene la logica di business per interagire con i servizi REST di JINA.
3. **DemoCA.Service.Infrastructure** (AuthenticationManager): Gestisce l'autenticazione e la comunicazione sicura con i servizi di JINA.

4.6. PROGRAM.CS (DEMOCA.CONSUMER)

Descrizione Generale

Il file Program.cs rappresenta il punto di ingresso dell'applicazione. Ha il compito di:

- Interpretare i parametri di esecuzione forniti dall'utente.
- Inizializzare l'ambiente e le configurazioni necessarie.
- Se l'ambiente non è produzione, offre la possibilità di inviare email di notifica con sostituzione del destinatario.
- Avviare le operazioni richieste tramite l'interfaccia ed interagisce con ActionManager.

Funzionalità Principali

Lettura dei Parametri input: Analizza gli argomenti della linea di comando o li recupera in modalità interattiva per determinare:

- l'ambiente (sviluppo, collaudo, produzione),
- l'istituzione (nel formati [IT:nnnnnn]),
- le modalità operative oggetto del documento (-cru, -clfu, -clau).

Per le modalità -clfu e -clau è prevista la password [superfil] per poter essere eseguita.

Verifica dei File di Input: recupera i file necessari per le operazioni di aggiunta o la cancellazione degli utenti se richiesti dai parametri.

Inizializzazione dell'Ambiente: Configura le variabili globali (istituzione, ambiente) e raccoglie le informazioni dall'ambiente scelto dalla tabella RINAEnvironment. Si legge la configurazione con IDInstitution uguale a istituzione ed Environment uguale ad ambiente. Carica la configurazione dall'ambiente selezionato, utilizzando la tabella

RINAEnvironment. Estrae le informazioni necessarie come Url, PortNumber, RESTServicePath, e le credenziali amministrative tramite join con la tabella Security.

Gestione del Flusso Operativo: In base ai parametri e alle modalità scelte, chiama i metodi appropriati di ActionManager per eseguire le operazioni richieste.

Dettagli Operativi

- **Metodo Main:**
  - **Analisi degli Argomenti**: Valuta gli argomenti passati all'applicazione e imposta le variabili globali.
  - **Scelta dell'Ambiente**: Richiede all'utente di scegliere l'ambiente di esecuzione e l'istituzione desiderata.
  - **Inizializzazione**: Imposta le configurazioni necessarie per l'esecuzione, tra cui il caricamento dei file per la gestione degli utenti.
  - **Esecuzione delle Operazioni**: Richiama i metodi per creare, eliminare o aggiornare utenti.
- **Gestione delle Modalità Operative**:
  - **-cru (Create Users)**: Avvia la procedura per aggiungere nuovi utenti nel sistema JINA chiamando CreateUser di ActionManager.
  - **-clfu (Clean Users from File)**: Avvia la procedura per eliminare utenti specificati in un file chiamando CleanUsers con la modalità appropriata.
  - **-clau (Clean All Users)**: Elimina tutti gli utenti recuperando la lista completa dal sistema RINA.

Interazione con l'Utente

- **Input Richiesto**:
  - Ambiente di esecuzione (sviluppo, collaudo, produzione).
  - Codice dell'istituzione nel formato [IT:nnnnnn].
  - Scelta se inviare notifiche email e, in caso, l'indirizzo email di sostituzione.
- **Feedback**:
  - Mostra messaggi di progresso e risultati delle operazioni.
  - Segnala eventuali errori o eccezioni durante l'esecuzione.

4.7 ACTIONMANAGER.CS (DEMOCA.SERVICE.WORKFLOW)

Descrizione Generale

La classe ActionManager contiene i metodi che implementano la logica di business per interagire con il sistema RINA. Utilizza chiamate HTTP per comunicare con i servizi REST esposti da RINA.

Funzionalità Principali

- **Gestione Utenti**:
  - Creazione di nuovi utenti (CreateUser).
  - Eliminazione di utenti (DeleteUsers).
  - Recupero della lista degli utenti (GetUsers).

Dettagli Operativi

CreateUser

- **Scopo**: Creare un nuovo utente nel sistema JINA con le specifiche fornite per ciascun utente presente nel file degli utenti.
- **Processo**:

1. **Recupero dei Gruppi**: Esegue una richiesta GET a eessiRest/Identity/Groups?institutionId=[istituzione] per ottenere i gruppi disponibili.
2. **Elaborazione dei Dati dell'utente**:
   - Genera la username utilizzando la 4ª colonna (la tronca dal carattere @).
   - Genera una password casuale.
   - Rimuove gli spazi vuoti nella 16ª colonna (gruppi).
   - Effettua un'inutile verifica di coerenza tra l'email composta e quella fornita.
3. **Associazione Gruppi e Ruoli**:
   - Analizza la colonna dei gruppi per associare i gruppi esistenti all'utente.
   - Verifica e assegna i ruoli in base alle colonne 10-15.
4. **Creazione dell'Utente**:
   - Crea un oggetto UserModel con i dati elaborati.
   - Invia una richiesta POST all'endpoint eessiRest/Identity/User con il modello dell'utente.
5. **Invio Email di Creazione**:
   - Se l'ambiente non è produzione e l'utente ha scelto di inviare email alternative, modifica il destinatario.
   - Invia l'email con le credenziali all'utente.

DeleteUsers

- **Scopo**: Eliminare utenti esistenti dal sistema RINA.
- **Processo**:

6. **Determinazione degli Utenti**: Riceve una lista di ID utente da eliminar
   - Per -clfu, legge gli ID utenti da un file.
   - Per -clau, recupera tutti gli utenti utilizzando la funzionalità **GetUsers**.

7. **Eliminazione**:
   - Per ogni ID utente, esegue una richiesta DELETE all'endpoint eessiRest/Identity/User/{Id utente}.

GetUsers

- **Scopo**: Recuperare la lista degli utenti associati a un'istituzione.
- **Processo**:

8. **Richiesta dei Dati**: Esegue una richiesta GET all'endpoint eessiRest/Identity/Users?institutionId={istituzione}.
9. **Filtraggio**: Utilizza espressioni regolari per filtrare gli utenti in base a specifici pattern nel campo Username.
10. **Restituzione**: Ritorna una lista di oggetti UserModel.

Gestione degli Errori e Retry

- **Retry**: Implementa meccanismi di ritentativo utilizzando RetryHelper, con un massimo di 5 tentativi e pause di 5 secondi tra i tentativi.
- **Eccezioni Personalizzate**: Utilizza eccezioni specifiche come RetryManagedException, WorkflowException e PolicyException per gestire errori noti.
- **Logging**: Registra informazioni di errore e di stato per facilitare il debugging e il monitoraggio.

4.8. AuthenticationManager (DemoCA.Service.Infrastructure)

Descrizione Generale

La classe AuthenticationManager è responsabile dell'autenticazione con il sistema RINA. Gestisce l'ottenimento dei ticket necessari per autenticare le richieste HTTP.

Funzionalità Principali

- **Ottenimento del Ticket di Grant (TGT)**: Attraverso GetTicketGrantingTicket, ottiene il TGT utilizzando le credenziali dell'utente.
- **Ottenimento del Ticket di Servizio (ST)**: Con GetServiceTicket, utilizza il TGT per ottenere l'ST necessario per il login.
- **Login e Impostazione degli Header**: Il metodo Login gestisce l'autenticazione finale e imposta gli header necessari per le richieste successive.

Dettagli Operativi

GetAuthenticatedClient

- **Scopo**: Restituire un HttpClient autenticato pronto per effettuare chiamate al sistema RINA.
- **Processo**:

11. **Inizializzazione del Client**: Crea un nuovo HttpClient e imposta la BaseAddress con l'URL e la porta dell'ambiente.
12. **Ottenimento del TGT**: Chiama GetTicketGrantingTicket passando il percorso del servizio REST e le credenziali.
13. **Ottenimento dell'ST**: Utilizza il TGT per ottenere l'ST con GetServiceTicket.
14. **Login**: Esegue il login tramite il metodo Login, che imposta anche gli header di autenticazione.
15. **Restituzione del Client**: Ritorna il HttpClient autenticato e pronto per l'uso.

GetTicketGrantingTicket

- **Processo**:

16. **Preparazione della Richiesta**: Crea un dizionario con le credenziali (username e password).
17. **Invio della Richiesta**: Invia una richiesta POST all'endpoint di autenticazione specificato.
18. **Estrazione del TGT**: Analizza la risposta per estrarre il TGT utilizzando espressioni regolari.

GetServiceTicket

- **Processo**:

19. **Preparazione della Richiesta**: Prepara i parametri necessari, inclusa la specifica del servizio.
20. **Invio della Richiesta**: Invia una richiesta POST all'endpoint dei ticket utilizzando il TGT.
21. **Ricezione dell'ST**: Ottiene l'ST dalla risposta della richiesta.

Login

- **Processo**:

22. **Impostazione degli Header**: Aggiunge l'header x-auth-cookie con il valore dell'ST.
23. **Esecuzione del Login**: Invia una richiesta GET all'endpoint di login con i parametri necessari.
24. **Impostazione del Token di Autenticazione**: Se la risposta ha successo, estrae il token X-XSRF-TOKEN e lo imposta negli header del client.

Considerazioni Aggiuntive

- **Gestione delle Credenziali**: Le credenziali utilizzate provengono dall'ambiente configurato (RINAEnvironment), che contiene informazioni sia per l'utente amministratore che per il supervisore.
- **Timeout**: Offre la possibilità di impostare un timeout infinito per gestire operazioni che potrebbero richiedere molto tempo.
- **Pulizia degli Header**: Dopo l'autenticazione, gli header precedenti vengono rimossi per evitare conflitti nelle richieste successive.

4.9 Flussi Operativi Dettagliati

Flusso di Creazione degli Utenti

2. **Avvio**: L'utente esegue l'applicazione con l'opzione -cru e specifica l'ambiente e l'istituzione.
3. **Lettura del File Utenti**: Program.cs legge il file di input contenente i dettagli degli utenti da creare. Il file è in formato testuale e ciascuna colonna è separata dal carattere di tabulazione. Ciascuna riga assume il seguente formato:

	Colonna	Note
1	Cognome	
2	Nome	
3	Non utilizzata	
4	Telefono	
5	Email	Convertito in minuscole
6	Non utilizzata	
7	Non utilizzata	
8	Non utilizzata	
9	Non utilizzata	
10	Viewer	Se vuoto non ha ruolo Viewer
11	Supervisor	Se vuoto non ha ruolo Supervisor
12	Authorized_Clerk	Se vuoto non ha ruolo Authorized_Clerk
13	Medical	Se vuoto non ha ruolo Medical
14	Unauthorized_Clerk	Se vuoto non ha ruolo Unauthorized_Clerk
15	Vip	Se vuoto non ha ruolo Vip
16	BUC/Sectors	Contiene una lista di coppie BUC/Sectors separate da virgola. Ciascuna coppia è formata da un BUC e da più Sector separati a loro volta dal carattere '/'. Da una singola coppia, per discriminare il BUC dai Sectors, si determina come BUC, il primo item che contiene la stringa '_BUC_'. A titolo esemplificativo la colonna: 03/04/05/06/XY_BUC_01,R_BUC_04/05/06/07,H_BUC_01/02a/02b/02c/03a/03b/05/06/07/10,M_BUC_03a/03b produce: XY {03, 04, 05, 06, 01}, R {04, 05, 06, 07}, H {02a, 02b, 02c, 03a, 03b, 05, 06, 07, 10}, M {03a, 03b}

4. **Autenticazione**: Viene ottenuto un HttpClient autenticato tramite AuthenticationManager.GetAuthenticatedClient.
5. **Elaborazione degli Utenti**:
    - Per ogni utente nel file:
        - ActionManager.CreateUser elabora i dati dell'utente.

- Recupera i gruppi associati all'istituzione.
- Assegna i ruoli a gruppi appropriati all'utente.

6. **Creazione nel Sistema RINA**:
   - Esegue una richiesta POST all'endpoint per la creazione degli utenti.
   - Verifica la risposta per assicurarsi che l'utente sia stato creato correttamente.
7. **Notifiche Email**:
   - Se configurato, invia una email di notifica all'utente con le credenziali e le istruzioni per l'accesso.

Flusso di Cancellazione degli Utenti

8. **Avvio**: L'utente esegue l'applicazione con l'opzione -clfu o -clau.
9. **Determinazione degli Utenti da Eliminare**:
   - Con -clfu, legge gli ID utenti da un file.
   - Con -clau, recupera tutti gli utenti associati all'istituzione.
10. **Autenticazione**: Viene ottenuto un HttpClient autenticato.
11. **Eliminazione**:
    - Per ogni ID utente, ActionManager.DeleteUsers invia una richiesta DELETE all'endpoint appropriato.
    - Gestisce eventuali errori e verifica l'esito dell'operazione.

4.10 Dettagli Tecnici Aggiuntivi

Database e Configurazioni

- **Tabella RINAEnvironment**:
  - Contiene le informazioni di configurazione per ogni ambiente.
  - Include dettagli come URL, numeri di porta, percorsi dei servizi e credenziali.
- **Tabella Security**:
  - Memorizza le credenziali degli utenti amministratori e supervisori.

Classi e Modelli

- **UserModel**:
  - Rappresenta il modello dei dati di un utente nel sistema RINA.
  - Include proprietà come Username, Password, FirstName, LastName, Email e Memberships.
- **Membership**:
  - Rappresenta l'associazione tra un utente, un gruppo e un ruolo.

4.11 Gestione delle Richieste HTTP

- Utilizza HttpClient per eseguire le chiamate ai servizi REST di JINA.

- Imposta gli header necessari per l'autenticazione e il tipo di contenuto.
- Gestisce le risposte e le eccezioni utilizzando meccanismi asincroni (async/await).

5. PROPOSTA PER NUOVA GESTIONE

5.1. OBIETTIVI NUOVA PROCEDURA

L'attuale processo di inserimento utenti richiede un significativo intervento manuale, aumentando il rischio di errori e rallentando l'operatività. Per migliorare l'efficienza e garantire l'accuratezza dei dati, è necessaria una nuova procedura che automatizzi il più possibile il processo.
Questa nuova soluzione avrà l'obiettivo di:

- **Eliminare o ridurre l'intervento umano** grazie a controlli automatici e validazioni dei dati.
- **Controllo dei dati**: validare e correggere errori automaticamente
- **Minimizzare gli errori** correggendo formattazioni errate e segnalando eventuali anomalie nei dati.
- **Velocizzare il processo** con un'integrazione diretta da file, database o API.
- **Garantire la tracciabilità** generando report dettagliati su utenti inseriti ed eventuali problemi riscontrati.
- **Interfaccia funzionale ed efficiente**

5.2. NUOVO WORKFLOW

Per garantire un processo più efficiente e conforme alle procedure aziendali, il nuovo workflow di gestione utenti seguirà una struttura chiara e automatizzata, con un passaggio formale per l'approvazione.
Il flusso di lavoro si articolerà nei seguenti passaggi:

1. **Tipologie di utenti**
2. **Emissione dell'Ordine di Servizio** → Ogni richiesta di creazione utente deve essere formalizzata tramite un **Ordine di Servizio.** Questo documento conterrà tutte le informazioni necessarie, inclusi nome, ruolo, permessi richiesti e motivazione dell'accesso.
3. **Raccolta e inserimento dei dati** → tramite inserimento diretto e/o tramite acquisizione file massivo
4. **Inserimento nel sistema** → Gli utenti validati verranno registrati nel sistema, con una gestione ottimizzata delle password, dei ruoli e delle autorizzazioni assegnate in base all'Ordine di Servizio.
5. **Validazione automatica** → Dati forniti in input dall'utente e/o file di input da caricamento massivo
6. **Report e notifiche** → Al termine del processo, verrà generato un **log dettagliato** con utenti creati, eventuali errori e suggerimenti per la correzione. Inoltre, verrà inviata una notifica agli uffici competenti per confermare l'attivazione dell'utente.

5.3. FLUSSO OPERATIVO COMPLESSIVO

1. **Autenticazione**:
   - L'utente accede alla procedura in base al proprio ruolo (Amministratore o Utente di Sede).
2. **Scelta dell'operazione**:
   - L'utente seleziona se effettuare un inserimento manuale, un inserimento tramite file massivo o eseguire operazioni di ricerca e gestione.
3. **Emissione dell'Ordine di Servizio**:
   - L'utente emette l'ordine di servizio che dovrà essere validato e allegato.

- L’operazione di inserimento, modifica o cancellazione sarà eseguita solo dopo l’approvazione dell’ordine di servizio.

4. **Inserimento dei Dati**:
   - Se è un **inserimento manuale**, il sistema verifica la matricola e i dati obbligatori.
   - Se è un **caricamento massivo**, il sistema controlla la validità dei dati prima del caricamento definitivo.
5. **Gestione delle Richieste**:
   - Le richieste di modifica e cancellazione, una volta confermate, saranno **pending** fino alla validazione dell’ordine di servizio.
6. **Generazione Log:**
   - al termine dell’operazione, sia con esito positivo sia negativo, verrà generato un log riassuntivo
7. **Ricerca e Report**:
   - L’utente esegue ricerche per identificare utenti o richieste, scarica i risultati e crea report statistici, quando necessario.

5.4. DETTAGLIO DEL FLUSSO

5.4.1 UTENTI ABILITATI

L’accesso alla procedura sarà riservato a due tipologie di utenti: l’Amministratore e l’Utente di Sede.

- L’Amministratore avrà la facoltà di caricare uno o più file per il caricamento massivo o, in alternativa, procedere con l’inserimento manuale;
- L’Utente di Sede, invece, potrà esclusivamente effettuare l’inserimento manuale.

5.4.2 EMISSIONE DELL'ORDINE DI SERVIZIO

L'emissione dell'ordine di servizio costituisce il vincolo imprescindibile per qualsiasi inserimento, modifica o cancellazione, sia tramite procedura standard che tramite caricamento massivo. Ogni ordine di servizio deve essere identificato da un numero di protocollo univoco, da registrare in una nuova tabella, e deve essere obbligatoriamente allegato affinché la procedura possa proseguire. Qualora il numero di protocollo risultasse già associato ad altre operazioni o non venisse allegato nessun documento, il sistema dovrà generare un messaggio di errore e non procedere con ulteriori funzioni.

5.4.3 INSERIMENTO MANUALE DEI DATI

L'inserimento manuale dei dati avverrà a partire dalla matricola dell’utente che si intende inserire, modificare o eliminare. La matricola rappresenta l’elemento chiave per l’identificazione dell’utente nel sistema e consentirà il recupero automatico dei relativi dati anagrafici. Una volta inserita la matricola, il sistema provvederà a verificare la presenza dell’utente nel Database Jira, controllando se esistano già registrazioni associate. Questa operazione è fondamentale per garantire la coerenza e l’integrità delle informazioni, evitando duplicazioni o incongruenze nei dati archiviati. Nel caso in cui l’utente risulti già presente nel database, potranno essere applicate solo le operazioni consentite dal sistema in base ai permessi dell’utente che effettua l’operazione. In assenza di una corrispondenza, sarà possibile procedere con un nuovo inserimento, assicurandosi che tutte le informazioni in merito alle autorizzazioni/accessi necessarie siano correttamente compilate prima della conferma definitiva.

5.4.4 INSERIMENTO CON FILE MASSIVO

Come precedentemente indicato, l'inserimento massivo dei dati sarà un'operazione riservata esclusivamente all'utente con ruolo di Amministratore. Per effettuare questa operazione, l'Amministratore avrà a disposizione due modalità distinte.

1. La prima modalità prevede la predisposizione manuale del file da caricare, che dovrà rispettare una specifica struttura a griglia. Tale formato garantirà la corretta organizzazione dei dati e permetterà di evitare errori di inserimento dovuti a incongruenze nel layout del file. L'utente Amministratore dovrà compilare i campi richiesti per ciascun nuovo utente da inserire, assicurandosi che tutte le informazioni obbligatorie siano presenti e conformi ai requisiti del sistema.

2. La seconda modalità, invece, permette un inserimento semplificato basato sull'inserimento delle singole matricole. In questo caso, l'Amministratore potrà richiedere che ai nuovi utenti vengano assegnate automaticamente le stesse autorizzazioni di un utente già esistente nel sistema. Questo meccanismo, assimilabile a una funzione di clonazione, consente di creare utenze gemelle, replicando i permessi e le configurazioni associate a un profilo di riferimento già approvato. Tale funzionalità risulta particolarmente utile per garantire coerenza nell'assegnazione dei permessi e velocizzare il processo di abilitazione di nuovi utenti con ruoli simili o identici all'interno del sistema.

Indipendentemente dalla modalità scelta, il sistema effettuerà controlli di validità sui dati inseriti prima di procedere all'effettivo caricamento nel database, segnalando eventuali errori o incongruenze che richiedano l'intervento dell'Amministratore per la correzione.

5.4.5 VALIDAZIONE AUTOMATICA DEL DATO

Indipendentemente dalla modalità di inserimento di una richiesta, sia essa effettuata manualmente o tramite caricamento massivo, il sistema eseguirà automaticamente una serie di controlli volti a garantire la congruità e la correttezza dei dati trasmessi. La natura e l'estensione di tali verifiche varieranno in base alla funzione specifica selezionata dall'utente, assicurando che ogni operazione venga eseguita nel rispetto delle regole di integrità e coerenza dei dati.

1. Nel caso di un nuovo inserimento, il primo controllo riguarderà la verifica della matricola dell'utente. Se la matricola risulta corretta, verranno caricati i dati anagrafici associati, evitando digitazioni manuali. Oltre alla matricola, alcuni dati saranno considerati obbligatori per la registrazione dell'utente, tra cui il tenant di appartenenza, lo stato dell'utente (attivo/inattivo) e tutte le autorizzazioni necessarie per garantirne il corretto funzionamento all'interno del sistema. L'assenza o l'errata compilazione di uno di questi elementi essenziali determinerà il blocco della procedura, con la segnalazione di un messaggio di errore che specificherà l'informazione mancante o non valida.
2. Nel caso di una modifica dei dati esistenti, il sistema effettuerà un confronto tra le informazioni aggiornate e quelle già presenti nel database. Verrà verificato che tutte le modifiche apportate rispettino le regole di dominio del dato, ovvero le restrizioni e i vincoli imposti per garantire la qualità e la consistenza delle informazioni archiviate. Qualsiasi variazione non conforme ai criteri stabiliti

comporterà il rifiuto dell'operazione, con l'indicazione dell'anomalia riscontrata affinché l'utente possa correggerla prima di procedere.

3. Per quanto riguarda la cancellazione di un utente, invece, il sistema non effettuerà controlli di congruità approfonditi, poiché l'operazione comporta semplicemente la rimozione dell'utente dal database. Tuttavia, per evitare eliminazioni accidentali, verrà sempre richiesta una conferma esplicita da parte dell'utente che sta eseguendo l'operazione. Solo dopo aver ricevuto tale conferma, il sistema procederà alla cancellazione definitiva dell'utente, assicurando così che l'azione sia deliberata e consapevole

5.4.6 GENERAZIONE LOG

Al termine di ogni operazione di inserimento, modifica o cancellazione, sia essa eseguita manualmente che tramite caricamento massivo, il sistema provvederà a generare un **log dettagliato** che registrerà l'esito dell'operazione. Questo log sarà essenziale per garantire la tracciabilità delle azioni effettuate, permettendo di monitorare in modo preciso tutte le operazioni compiute all'interno del sistema.
Il **log** conterrà informazioni fondamentali come il **tipo di operazione** (inserimento, modifica, cancellazione), il **numero di protocollo** dell'ordine di servizio associato, l'**utente** che ha effettuato l'operazione, la **data e ora** di esecuzione e, soprattutto, l'esito dell'operazione, che sarà registrato come **positivo** o **negativo**. In caso di esito negativo, verrà specificato anche il **motivo del fallimento**, indicando se l'operazione non è stata eseguita a causa di errori nei dati, problemi di congruità o mancata conferma dell'ordine di servizio.
Per ogni operazione, sia manuale che tramite **caricamento massivo**, il sistema assicurerà la corretta registrazione e la **conservazione sicura** del log, in modo che ogni evento possa essere successivamente consultato per verificare l'accuratezza dei dati e la corretta esecuzione dei processi. Inoltre, il log sarà strutturato in modo da permettere una facile consultazione e ricerca, con la possibilità di filtrare i dati per tipo di operazione, esito o periodo di tempo, in modo da facilitare l'analisi e l'individuazione di eventuali problematiche o anomalie.
Il sistema di logging avrà inoltre una funzione di **allerta** che notificherà automaticamente agli amministratori in caso di errori gravi o anomalie ricorrenti, consentendo un intervento tempestivo per risolvere eventuali criticità. Questo processo garantirà non solo il **monitoraggio delle attività** svolte, ma anche un livello superiore di **sicurezza e controllo**, in quanto ogni operazione sarà pienamente tracciata e documentata, permettendo un'analisi storica completa delle modifiche e delle interazioni con il sistema.

5.4.1 FUNZIONI DI RICERCA E STATISTICA

La procedura includerà funzionalità di ricerca, consentendo agli utenti di individuare rapidamente informazioni attraverso una serie di filtri personalizzabili. Sarà possibile applicare diversi criteri di selezione, come matricola, nome e cognome, stato dell'utente, sede di appartenenza, settore o autorizzazioni assegnate. Questo permetterà di ottenere risultati mirati in base alle esigenze operative dell'utente.
I risultati delle ricerche potranno essere gestiti in due modalità principali. Da un lato, sarà possibile scaricare l'elenco completo delle informazioni trovate in un file Excel, facilitando così l'archiviazione, l'analisi offline o l'integrazione con altri strumenti aziendali. Dall'altro, le singole voci restituite dalla ricerca potranno essere selezionate per generare direttamente richieste di modifica o cancellazione. Tali richieste rimarranno in stato 'pending' fino alla ricezione e validazione dell'ordine di servizio, il quale fungerà da elemento autorizzativo per l'effettiva esecuzione dell'operazione.

L’accesso a queste funzionalità di ricerca e gestione sarà garantito a tutte le categorie di utenti, sia agli Amministratori sia agli Utenti di Sede, affinché possano effettuare le operazioni in base ai rispettivi ruoli e necessità.
Oltre alle funzioni di ricerca e gestione delle richieste, la procedura consentirà anche l’estrazione di dati statistici. Sarà possibile, ad esempio, ottenere informazioni aggregate come il numero totale di utenti autorizzati per sede, il conteggio degli utenti suddivisi per settore, l’analisi della distribuzione delle autorizzazioni concesse e altre metriche rilevanti per il monitoraggio e l’ottimizzazione del sistema. Questi dati potranno essere esportati in file scaricabili e utilizzati per la creazione di report personalizzati, fornendo così strumenti utili per il controllo e la pianificazione delle risorse aziendali

5.5. DATABASE

Attualmente, la gestione degli utenti è affidata alla tabella **iam_user** e alle sue tabelle correlate, al fine dell’l’inserimento, la modifica e la cancellazione degli utenti queste sono le relazioni principali:

Relazioni in ingresso (tabelle che fanno riferimento a iam_user)

Le seguenti tabelle contengono una **foreign key** che punta a **iam_user**:

- **iam_user_group**
- iam_group
- **iam_role**

Relazioni in uscita (tabelle a cui iam_user fa riferimento)

- **tenant**

Con la nuova Gestione Utenti, verrà creata una versione centralizzata speculare delle tre tabelle (**iam_user**, **iam_group**, **iam_role**) che conterrà i dati degli utenti di tutti i Tenant. Sarà previsto un **allineamento iniziale una tantum** per sincronizzare i dati esistenti, ed inoltre per garantire la **coerenza dei dati**, sarà implementato un **batch serale periodico** che verificherà l'allineamento tra le tabelle centralizzate e quelle già presenti nel database **EESSI-RINA**.

Le nuove tabelle si chiameranno NGUJ_USER, NGUJ_GROUP e NGUJ_ROLE, e tutte le colonne avranno il prefisso NGUJ, che sta per Nuova Gestione Utenti Jina.

Le nuove tabelle avranno il compito di **centralizzare le informazioni** provenienti da tutte le direzioni, evitando **duplicazioni** e riducendo il rischio di errori di inserimento. La nuova procedura prevederà un doppio colloquio: uno sulle nuove tabelle centralizzate per leggere, salvare e aggiornare i dati, e un altro tramite i servizi già esistenti per l'accesso e l'aggiornamento del database **EESSI-RINA PostgreSQL**.
Per gestire il colloquio e la completezza dei dati alle 3 tabelle verranno inserire/eliminate alcune colonne rispetto a quelle presenti nel Database EESSI-RINA, se ne evidenziano le differenze di seguito (le colonne nuove sono evidenziate in giallo):

tabella NGUJ_USER		
Nome colonna	Tipo dati	Commento
NGUJ_SID	INT8	The surrogate key
NGUJ_TENANT	VARCHAR(255)	Tenant
NGUJ_FK_GROUP_SID	INT8	Foreign key to the group table
NGUJ_FK_ROLE_SID	INT8	Foreign key to the role table
NGUJ_USERNAME	VARCHAR(255)	The username of the User
NGUJ_ID	VARCHAR(255)	The id of the iam user.
NGUJ_FIRST_NAME	VARCHAR(255)	The first name of the User
NGUJ_LAST_NAME	VARCHAR(255)	The last name of the User
NGUJ_MIDDLE_NAMES	VARCHAR(255)	The middle names of the User
NGUJ_PHONE_NUMBER	VARCHAR(255)	
NGUJ_EMAIL	VARCHAR(255)	The email address of the User
NGUJ_KEYSTORE_ALIAS	VARCHAR(1024)	
NGUJ_PASSWORD	VARCHAR(255)	The password of the User
NGUJ_SALT	VARCHAR(255)	Salt field used for encrypting the password
NGUJ_IS_SYSTEM	BOOL	The is system flag.
NGUJ_IS_ENABLED	BOOL	True is the user is enabled
NGUJ_IS_DELETED	BOOL	Flag indicating if this user is deleted
NGUJ_IS_ADMIN	BOOL	Flag indicating if this user is an administrator
NGUJ_CREATED_AT	TIMESTAMP WITH TIME ZONE	The time that the user was first inserted into the DB
NGUJ_UPDATED_AT	TIMESTAMP WITH TIME ZONE	The last time that the user was updated
NGUJ_CREATED_BY	VARCHAR(255)	The person/process that created the record.
NGUJ_UPDATED_BY	VARCHAR(255)	The person/process that last updated the record.

tabella NGUJ_GROUP		
Nome colonna	Tipo dati	Commento
NGUJ_SID	INT8	the surrogate key
NGUJ_NAME	VARCHAR(20)	The role name

tabella NGUJ_GROUP		
Nome colonna	Tipo dati	Commento
NGUJ_SID	INT8	the surrogate key
NGUJ_FK_ROLE_SID	INT8	Foreign key to the role table
NGUJ_TENANT	VARCHAR(255)	Tenant
NGUJ_NAME	VARCHAR(255)	The name of the group
NGUJ_ID	VARCHAR(255)	The id.
NGUJ_DISPLAY_NAME	VARCHAR(255)	The display name of the Group
NGUJ_PARENT_PATH	VARCHAR(1024)	The path to the parent.
NGUJ_DESCRIPTION	VARCHAR(255)	The description of the Group
NGUJ_IS_DELETED	BOOL	Flag indicating if this group is deleted
NGUJ_IS_ORGANISATION_UNIT	BOOL	The is organisation unit flag.
NGUJ_CREATED_AT	TIMESTAMP WITH TIME ZONE	The time that the group was first inserted into the DB
NGUJ_UPDATED_AT	TIMESTAMP WITH TIME ZONE	The last time that the group was updated
NGUJ_CREATED_BY	VARCHAR(255)	The person/process that created the record.
NGUJ_UPDATED_BY	VARCHAR(255)	The person/process that last updated the record.